EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Alan Davis (reg # 39,954) on 3/13/2008.

The application has been amended as follows:

1. (Currently Amended) An integrated circuit functioning as a network interface adapter, comprising:

a plurality of media access controllers (MACs), for transceiving packets;

a local bus interface, for performing addressed data transfers on a local bus coupled thereto;

a bus router, for performing transport layer operations between said plurality of MACs and said local bus interface;

a memory, shared by said plurality of MACs, said local bus interface, and said bus router, for buffering data received thereby; and

a transaction switch, coupled to [[each of]] said memory, said plurality of MACs, said local bus interface, and said bus router, for switching data and transactions therebetween;

wherein said bus router is configured to write a packet header into said memory via said transaction switch along with addressed data stored in said memory by said local bus interface to create a packet;

wherein said local bus interface is configured to read a payload portion of a packet stored in said memory and to transmit said payload portion on said local bus coupled thereto;

wherein said payload portion is located in said memory at an offset specified in a transaction posted by said bus router to said local bus interface via said transaction switch.

6-8. (Canceled)

23. (Currently Amended) A transaction switch for switching data between a plurality of data devices, comprising:

a memory, shared by the plurality of data devices for buffering data received thereby;

multiplexing logic, coupled to said memory, for controlling the transfer of data

    between the plurality of data devices and said memory; and

control logic, coupled to said multiplexing logic, for controlling said multiplexing

    logic;

wherein the plurality of data devices comprise a plurality of packetized data

    devices and a plurality of addressed data devices;

wherein said control logic is configured to selectively control said multiplexing

    logic to transfer data through said memory between two of said packetized

    data devices and between one of said packetized data devices and one of

    said addressed data devices;

wherein said control logic is configured to selectively control said multiplexing

    logic to transfer data through said memory between two of said packetized

    data devices and between one of said packetized data devices and one of

    said addressed data devices in response to a transaction posted to the

    transaction switch by the plurality of data devices;

wherein said transaction comprises a command to transfer data between said

    memory and one of the plurality of data devices;

wherein said transaction comprises an address of a buffer within said memory,

    wherein said data to be transferred in response to said command is stored

    in said memory.

30-32. (Canceled)

33. (Currently Amended) The transaction switch of claim [[32]]23, wherein said

    transaction comprises an offset within said buffer for addressing portions of said

    data.

34. (Currently Amended)   The transaction switch of claim [[30]]23, wherein said

    transaction comprises a tag for uniquely identifying said transaction from other

    transactions posted to the transaction switch by the plurality of data devices.

36. (Currently Amended)   A transaction switch for switching transactions and data

    between a plurality of data interfaces, the transaction switch comprising:

    a memory, shared by the plurality of data interfaces, for buffering data received

        thereby;

    a plurality of transaction queues, associated with each of the plurality of data

        interfaces, configured to store transactions, said transactions adapted to

        convey information to enable the plurality of data interfaces to transfer

        said data according to a plurality of disparate data transfer protocols

        supported thereby; and

    control logic, coupled to said memory and said plurality of transaction queues,

        configured to route said data through said shared memory between the

plurality of data interfaces and to switch said transactions between the

plurality of data interfaces;

wherein at least one of the plurality of data interfaces comprises a bus router for

performing a transport layer function between at least two of the plurality

of data interfaces which support disparate data protocols;

wherein at least one of the plurality of data interfaces comprises a local bus

interface for interfacing to a local bus, wherein at least one of the plurality

of data interfaces comprises a MAC for interfacing to a network, wherein

said bus router is configured to perform said transport layer function by

writing packet header information into an allocated portion of said memory

in front of data written into said allocated portion of said memory by said

local bus interface;

wherein said bus router is configured to write said packet header information into

said allocated portion of said memory before said MAC reads said packet

from said allocated portion of said memory for writing to said network,

without copying said data to another memory.

41. (Currently Amended)   An integrated circuit, comprising:

at least three data interfaces;

a memory, shared by said at least three data interfaces for buffering data

therebetween; and

a transaction switch, coupled to said at least three data interfaces and said

memory, for dynamically allocating portions of said memory to said at

least three data interfaces for storing data therein, and for controlling

access to said allocated portions of said memory by each of said at least

three data interfaces;

wherein at least one of said at least three data interfaces is of a different type

than the others;

wherein at least one of said at least three data interfaces comprises a bus router

for performing a transport layer function between at least two other of said

at least three data interfaces which support disparate data protocols;

wherein at least one of said at least three data interfaces comprises a local bus

interface for interfacing to a local bus, wherein at least one of said at least

three data interfaces comprises a MAC for interfacing to a network,

wherein said bus router is configured to perform said transport layer

function by writing packet header information into said allocated portions

of said memory in front of data written into said allocated portions of said

memory by said local bus interface;

wherein said bus router is configured to write said packet header information into

said allocated portions of said memory before said MAC reads said packet

from said allocated portion of said memory for writing to said network,

without copying said data to another memory.

57-59. (Canceled)

63-66. (Canceled)

67. (Currently Amended)   A network interface adapter, comprising:

a plurality of network ports, for interfacing to a network;

at least one addressed data bus interface;

a memory, for buffering data received by said plurality of network ports and said
at least one addressed data bus interface;

a transport layer engine for routing said data between said plurality of network
ports and said at least one addressed data bus interface;

a plurality of transaction queues, associated with each of said plurality of network
ports, said at least one addressed data bus interface and said transport
layer engine, for storing transactions; and

a transaction switch, coupled to said plurality of transaction queues, configured to
route said transactions between said plurality of network ports, said at
least one addressed data bus interface and said transport layer engine;

a bus router for performing a transport layer function between said at least one
addressed data bus interface and said plurality of network ports;

wherein said bus router is configured to perform said transport layer function by

writing packet header information into an allocated portion of said memory

in front of data written into said allocated portion of said memory by said at

least one addressed data bus interface;

wherein said bus router is configured to write said packet header information into

said allocated portion of said memory before one of said plurality of

network ports reads said packet from said allocated portion of said

memory for writing to said network, without copying said data to another

memory.

71-78. (Canceled)

79. (Currently Amended)   A transaction switch in a network device having a buffer

memory and plurality of data devices, including packetized and addressed data

devices, the transaction switch comprising:

a buffer manager, for allocating portions of the buffer memory to the plurality of

data devices on an as-needed basis;

a plurality of data paths, coupling the buffer memory and the plurality of

packetized and addressed data devices, for providing the plurality of data

devices access to the buffer memory;

a mapping table, for storing packet destination identification information;

a plurality of transaction queues, for transferring transactions between the

transaction switch and the plurality of data devices; and

control logic, coupled to said mapping table and said plurality of transaction

queues, for selectively switching transactions between the plurality of data

devices based on said mapping table information and based on contents

of said transactions;

wherein at least one of the plurality of data interfaces comprises a bus router for

performing a transport layer function between at least two of the plurality

of data interfaces which support disparate data protocols;

wherein at least one of the plurality of data interfaces comprises a local bus

interface for interfacing to a local bus, wherein at least one of the plurality

of data interfaces comprises a MAC for interfacing to a network, wherein

said bus router is configured to perform said transport layer function by

writing packet header information into said allocated portions of the buffer

memory in front of data written into said allocated portions of the buffer

memory by said local bus interface;

wherein said bus router is configured to write said packet header information into

said allocated portions of the buffer memory before said MAC reads said

packet from said allocated portion of the buffer memory for writing to said

network, without copying said data to another memory.

88-92. (Canceled)

97-99. (Canceled)

100. (Currently Amended)   A network interface adapter, comprising:

    a local bus interface, configured to connect the adapter to a local bus and to
        perform an addressed data protocol on the local bus;

    a media access controller (MAC), configured to connect the adapter to a network
        and to perform a packetized data protocol on the network;

    a transport protocol engine, configured to perform protocol translation from the
        addressed data protocol to the packetized data protocol by creating a
        packet from data received on the local bus by the local bus interface;

    a memory, directly accessible by each of the local bus interface, the MAC, and
        the transport protocol engine in a random access fashion;

    a transaction switch, coupled to the local bus interface, the MAC, the transport
        protocol engine, and the memory, and configured to dynamically allocate a
        portion of the memory for buffering the data and the packet;

    wherein the transport protocol engine is configured to perform the protocol
        translation before the MAC reads the packet from the memory for
        transmission on the network, wherein the transport protocol engine

performs the protocol translation without copying the data to another

memory;

wherein the transport protocol engine is further configured to perform second

protocol translation from the packetized data protocol to the addressed

data protocol by identifying a data payload within a second packet

received from the network by the MAC and specifying a local bus address

for the data payload;

wherein the transport protocol engine is further configured to perform the second

protocol translation before the local bus interface reads the data payload

from the memory for writing on the local bus to the local bus address,

wherein the transport protocol engine performs the second protocol

translation within the memory without copying the data to another

memory.

101. (Canceled)

104. (Currently Amended)   A network interface adapter, comprising:

a local bus interface, configured to connect the adapter to a local bus and to

perform an addressed data protocol on the local bus;

a media access controller (MAC), configured to connect the adapter to a network

and to perform a packetized data protocol on the network;

a transport protocol engine, configured to perform protocol translation from the

packetized data protocol to the addressed data protocol by identifying a

data payload within a packet received from the network by the MAC and

specifying a local bus address for said data payload;

a memory, directly accessible by each of the local bus interface, the MAC, and

the transport protocol engine in a random access fashion;

a transaction switch, coupled to the local bus interface, the MAC, the transport

protocol engine, and the memory, and configured to dynamically allocate a

portion of the memory for buffering the data payload and the packet;

wherein the transport protocol engine is configured to perform the protocol

translation before the local bus interface reads the data payload from the

memory for writing on the local bus to said local bus address, wherein the

transport protocol engine performs the protocol translation without copying

the data to another memory;

wherein the transport protocol engine is further configured to perform second

protocol translation from the addressed data protocol to the packetized

data protocol by creating a second packet from data received on the local

bus by the local bus interface;

wherein the transport protocol engine is further configured to perform the second

protocol translation before the MAC reads the second packet from the

memory for transmission on the network, wherein the transport protocol

engine performs the second protocol translation without copying the data

to another memory.

105. (Canceled)

108. (Currently Amended)   A method for performing a transport layer function without

double-buffering in a network interface adapter comprising a local bus interface,

a media access controller (MAC), a transport protocol engine, and a memory

managed by the transaction switch and directly accessible by each of the local

bus interface, the MAC, and the transport protocol engine in a random access

fashion, the method comprising:

allocating, by the transaction switch, a portion of the memory for buffering data

received on the local bus by the local bus interface;

writing, by the local bus interface, the data into the allocated portion of the

memory;

creating, by the transport protocol engine, with the data a packet within the

allocated portion of the memory;

reading, by the MAC, the packet from the allocated portion of the memory for

transmission on a network, after said creating, wherein the transport

protocol engine performs said creating the packet without copying the data

to another memory;

allocating, by the transaction switch, a second portion of the memory for buffering

a second packet received from the network by the MAC;

writing, by the MAC, the second packet into the second allocated portion of the

memory;

identifying, by the transport protocol engine, a data payload within the second

packet;

specifying, by the transport protocol engine, a local bus address for the data

payload; and

reading, by the local bus interface, the data payload from the second allocated

portion of the memory for writing on the local bus to the local bus address,

after said identifying and said specifying, wherein the transport protocol

engine performs said identifying and said specifying without copying the

data to another memory.

109. (Canceled)

111. (Currently Amended)   A method for performing a transport layer function without

double-buffering in a network interface adapter comprising a local bus interface,

a media access controller (MAC), a transport protocol engine, and a memory

managed by a transaction switch and directly accessible by each of the local bus

interface, the MAC, and the transport protocol engine in a random access
fashion, the method comprising:

allocating, by the transaction switch, a portion of the memory for buffering a
packet received from the network by the MAC;

writing, by the MAC, the packet into the allocated portion of the memory;

identifying, by the transport protocol engine, a data payload within the packet;

specifying, by the transport protocol engine, a local bus address for the data
payload;

reading, by the local bus interface, the data payload from the allocated portion of
the memory for writing on the local bus to the local bus address, after said
identifying and said specifying, wherein the transport protocol engine
performs said identifying and said specifying without copying the data to
another memory;

allocating, by the transaction switch, a second portion of the memory for buffering
data received on the local bus by the local bus interface;

writing, by the local bus interface, the data into the second allocated portion of
the memory;

creating, by the transport protocol engine, with the data a second packet within
the second allocated portion of the memory; and

reading, by the MAC, the second packet from the second allocated portion of the

memory for transmission on a network, after said creating, wherein the

transport protocol engine performs said creating the second packet

without copying the data to another memory.

112. (Canceled)

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to DUYEN M. DOAN whose telephone number is

(571)272-4226. The examiner can normally be reached on 9:30am-6:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Bunjob Jaroenchonwanit can be reached on (571) 272-3913. The fax

phone number for the organization where this application or proceeding is assigned is

571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/D. M. D./

Examiner, Art Unit 2152

/Bunjob Jaroenchonwanit/

Supervisory Patent Examiner, Art Unit 2152